ESIEE Paris
IGI-3008

Projet de programmation

Liste des thèmes

1 Aide à l'arbitre d'Ultimate Ping 3
2 2048 3
3 Calendrier 4
4 Convertisseur d'unités 4
5 Motus 5
6 Les taquins 5
7 Explorateur de l'ensemble de Mandelbrot 6
8 Cryptographie 6
9 Solveur de problème d'échecs : mat en deux coups 7
10 Aide grammaticale 7
11 Recherches d'itinéraires en transport en commun 7
12 Jeu de la vie 8

Année scolaire 2015-2016

1 Modalités

Le projet se fera par groupes de deux à trois élèves issus du même groupe de TP. C'est un travail original qui est demandé : une copie ou une inspiration trop marquée de programmes existants sera sanctionnée par un F.

La programmation se fera impérativement en Python. Une version *console* (avec une interface utilisateur avec `print` et `input`) sera écrite avant toute tentative de réaliser une interface graphique.

Dans un premier temps, après une recherche documentaire sur le thème traité, on réfléchira, et ce, sans écrire une seule ligne de programme, à l'apparence du produit final :

— que va-t-on proposer à l'utilisateur ? (les objectifs)
— comment va-t-on lui proposer ?

et l'on décrira complètement une session (spécifications des interfaces utilisateur) :

— quels seront les différents écrans qui apparaîtront à l'utilisateur (au lancement, en cours de session, ...) ?
— pour chacun de ces écrans, quelles seront les possibilités d'action de l'utilisateur ?
— pour chacun des choix de l'utilisateur, comment lui apparaîtront les résultats ?

Puis, on découpera le travail à faire en modules les plus indépendants possibles les uns des autres, tout en veillant à la cohérence globale du projet. Les spécifications des fonctions et/ou des classes à écrire seront alors rédigées. La rédaction d'un premier rapport clôturera cette phase de travail.

La seconde phase consistera à écrire, tester, documenter et lier les différents morceaux de programme jusqu'au produit final, puis à rédiger le guide utilisateur (mode d'emploi, destiné à l'utilisateur final) et le guide de maintenance (documentation technique, destinée à la maintenance du produit).

2 Documents à remettre

Les documents (sauf les listings) seront obligatoirement rédigés sur traitement de texte.
La forme et le fond seront évaluées et contribueront à le note finale.

1. pour le 6 novembre 2015 un rapport d'analyse décrivant :
   — les objectifs ;
   — les spécifications des interfaces utilisateurs ;
   — le découpage du programme en modules ;
   — les spécifications des traitements à écrire.
2. pour le 17 décembre 2015
   — les listings commentés des programmes (attention aux mises en page automatiques des outils bureautiques, inérerz vos programmes tels quels) ;
   — les jeux de tests auxquels votre programme a été soumis ;
   — un guide utilisateur ;
   — un guide de maintenance ;
   — un rapport sur les écarts entre objectif et réalisation, et les raisons de ces écarts.

3 Soutenances (début janvier 2016)

Une soutenance clôturera ce travail. Elle durera une demi-heure et nécessitera la présence du groupe de projet au complet. Elle sera constituée d'une démonstration d'une dizaine de minutes, puis d'une interrogation sur le projet. Chaque membre du groupe devra être capable d'expliquer toute partie du programme.

4 Thèmes des projets

Thème 1 Aide à l'arbitre d'Ultimate Ping

Le but de ce projet est de réaliser un programme informatique permettant de mettre à jour le score d'un match d'Ultimate Ping. Il faut bien entendu se renseigner sur les règles de comptage des points de ce sport.

Le programme devra demander à l'utilisateur de rentrer le nom des joueurs. Puis il proposera un choix entre soit une nouvelle partie (initialisation du score à 0-0), soit reprendre une partie (demande du score actuel à l'utilisateur). Dans les deux cas, le programme demandera le nom du joueur au service.

Puis, après l'entrée de chaque point marqué par un joueur, le programme affichera automatiquement le score ainsi que le nom du joueur au service.

Une interface console pourra être la suivante :

```
              Service Set 1   Set 2   Set 3   Jeu
1. Dupont        *      4       2       0      2
2. Durand               1       4       1      1

Entrez le numéro du joueur qui a marqué le point
(0 pour annuler la dernière entrée) : _
```

Facultativement, le programme pourra être paramétré pour d'autres sports avec règles de comptage de points similaires (tennis, tennis de table, volley-ball, ...).

Thème 2 2048

Le but de ce projet est d'écrire un programme permettant à l'utilisateur de jouer à 2048. Si vous faites partie des deux mille quarante-huit être humains qui ne connaissent pas ce jeu, voir la page Wikipedia 2048.

Une interface console pourra être la suivante :

```
------------------------------
Score : 180

  -     2     4     8
  -     4     8     2
  -     2     4     8
  -     2     4    32
------------------------------
Coup (E,N,W,S) ? e
Coup impossible
Coup (E,N,W,S) ? s
------------------------------
Score : 192

  -     -     -     8
  -     2     4     2
  2     4     8     8
  -     4     8    32
------------------------------
Coup (E,N,W,S) ?
```

Facultativement, le programme pourra donner des conseils ou même jouer automatiquement.

Thème 3 Calendrier

Le but de ce projet est de réaliser un calendrier informatique. Le calendrier sur lequel reposera le programme sera le calendrier en usage aujourd'hui en France (dit calendrier Grégorien).

Les possibilités suivantes devront obligatoirement être disponibles pour l'utilisateur du programme final :

- visualiser le calendrier d'un mois, d'une année ;
- visualiser le nom du jour d'une date donnée ;
- visualiser le nombre de jours séparant deux dates.

Optionnellement, on pourra réfléchir et réaliser une ou plusieurs des possibilités suivantes :

- autres calendriers (chinois, israélite, musulman, républicain, ...) ;
- ajout à une date d'un certain nombre de jours, semaines et/ou mois ;
- calcul des phases lunaires, des jours fériés ;
- réponse à des interrogations du type : prochain vendredi 13, mois à 6 semaines.

L'affichage d'un mois pourra être le suivant :

```
      septembre 2015
lun mar mer jeu ven sam dim
      1   2   3   4   5   6
  7   8   9  10  11  12 *13*
 14  15  16  17  18  19  20
 21  22  23  24  25  26  27
 28  29  30
```

Voir aussi cette applet (on ne vous en demande pas tant).

Important : ce projet doit être écrit en Python de base, sans importation de modules extérieurs comme `calendar` ou `locale`.

Thème 4 Convertisseur d'unités

Le but de ce projet est de réaliser un programme aidant l'ingénieur à convertir la valeur d'une mesure faite dans une unité en sa valeur dans une autre unité (par exemple, la conversion de 100 °F en °C ou 3250 m en km). Un ou plusieurs fichiers textes éditables par l'utilisateur contiendront les unités disponibles et les formules de conversion. Le programme exploitera ces données pour répondre aux demandes de l'utilisateur.

Les possibilités suivantes devront obligatoirement être disponibles pour l'utilisateur du programme final :

- conversion dans les deux sens des unités françaises et anglo-saxonnes pour les longueurs, vitesses, poids et températures ;
- conversion à l'intérieur d'un système entre multiples et sous-multiples ;
- affichage des formules de conversions.

Toute autre possibilité du logiciel dans le domaine de la conversion de mesure sera la bienvenue : écriture anglo-saxonne des nombres (1,234.56) pour les unités anglo-saxonnes, écriture française (1 234,56) pour les unités françaises, anciennes unités de mesure, etc.)

Une interface console pourrait être la suivante :

```
Entrez une conversion à effectuer : 12.5 km->mile
12.5 km = 7.767 miles
Entrez une conversion à effectuer :
```

Important : ce projet doit être écrit en minimisant les duplications de code.

Thème 5 Motus

Le but de ce projet est de permettre à l'utilisateur de jouer à la phase *Lettres* du jeu télévisé Motus :

Un mot de 7 lettres est choisi dans un dictionnaire.

La première lettre est révélée au joueur.

L'objectif du joueur est de deviner ce mot en faisant des propositions.

Si le mot proposé par le joueur contient des lettres communes avec le mot à deviner, ces lettre sont marquées (un carré rouge entoure une lettre à la bonne place, un rond jaune entoure une lettre présente, mais mal placée).

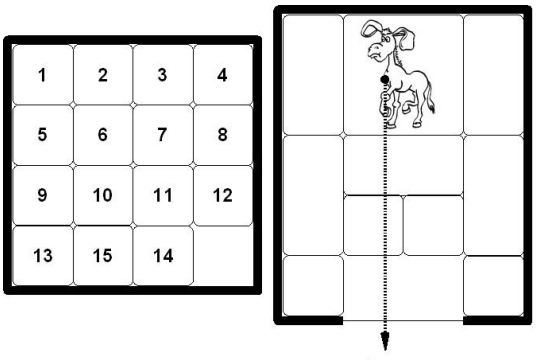
Une proposition incorrecte (mot trop court, trop long ou absent du dictionnaire) n'est pas évaluée.

Une interface console pourrait être la suivante :

```
A------
            Votre proposition ? amateur
AMATEUR
x-----o
            Votre proposition ? arrondi
AMATEUR
x-----o
ARRONDI
xo-oo-o
            Votre proposition ? aggrave
AMATEUR
x-----o
ARRONDI
xo-oo-o
AGGRAVE
x--x-o-
```

Facultativement, le programme pourra essayer de deviner un mot choisi par le joueur.

Thème 6 Les taquins



Le 15-14 L'Âne rouge

Le but de ce projet est de réaliser une version informatique d'un jeu de taquin.

Un taquin est un jeu composé de plaques que l'on fait glisser une par une dans des espaces de manière à les faire passer d'une disposition initiale à une disposition finale.

Par exemple, il s'agissait dans le célèbre *15-14* de Sam Loyd de remettre les plaques dans l'ordre (impossible) et dans l'*Âne Rouge* de faire parvenir la grande plaque carrée au-dessus de l'ouverture du bas.

On axera essentiellement la réalisation sur la jouabilité et la possibilité de créer facilement de nouveaux taquins.

Thème 7 Explorateur de l'ensemble de Mandelbrot

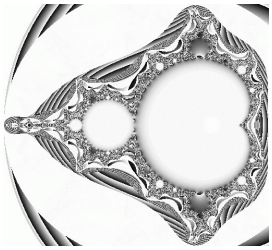
Le but de ce projet est de créer une cartographie de l'ensemble de Mandelbrot, en autorisant les zooms et les redéfinitions des couleurs de régions.

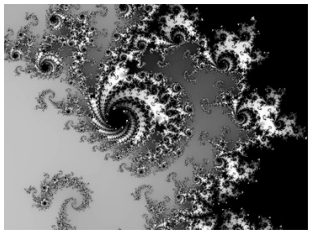
L'ensemble de Mandelbrot est l'ensemble des nombres complexes c tels que la suite définie par $z_0 = 0$ et $z_{n+1} = z_n^2 + c$ ait tous ses éléments bornées en norme. On démontre que si un des éléments de la suite a une norme supérieure à 2, elle diverge.

L'ensemble de Mandelbrot est représenté par l'ensemble des images de ces nombres dans le plan, coloriées suivant le nombre d'itérations nécessaires à la divergence.

Les possibilités suivantes devront obligatoirement être disponibles pour l'utilisateur du programme final :

- choisir les dimensions d'une fenêtre d'affichage ;
- choisir les bornes du sous-ensemble visualisé ;
- choisir les couleurs et les limites de divergence.





Facultativement, on pourra réfléchir et réaliser les fonctionnalités suivantes :

- sauvegarder les images intéressantes, avec leurs paramètres de tracé et les restaurer ;
- effectuer des zooms avant/arrière sur une image.

Thème 8 Cryptographie

Le but de ce projet est d'écrire une application permettant de coder (et de décoder) des messages cryptés.

Trois principes de cryptage seront obligatoirement intégrés au produit :

- le chiffre de César, voir https://fr.wikipedia.org/wiki/Chiffrement_par_décalage
- le chiffre de Vigenère, voir https://fr.wikipedia.org/wiki/Chiffre_de_Vigenère
- la machine Enigma, voir https://fr.wikipedia.org/wiki/Enigma_(machine)

Le programme devra permettre de choisir une opération (codage, décodage), un principe de codage, une clef de chiffrage et un texte. Il devra fournir le texte crypté (décrypté) correspondant.

Facultativement, il pourra tenter de décrypter une message crypté sans clé (combinatoire, analyse de fréquences, indices de coïncidence). Facultativement, étant donné un message crypté, il pourra essayer de retrouver la configuration de la machine et le décrypter.

Thème 9 Solveur de problème d'échecs : mat en deux coups

Le but de ce projet est de réaliser un programme résolvant les problèmes de mats en deux coups aux échecs.

Les possibilités suivantes devront obligatoirement être disponibles pour l'utilisateur du programme final :

- mise en place manuelle des pièces, avec vérification des positions illégales ;
- mise en place des pièces à partir d'une base de problèmes éditables par l'utilisateur ;
- choix du camp tenu par le programme ;

Le programme devra afficher le coup à jouer pour donner échec et mat au camp adverse en deux coups contre toute défense (ou indiquer l'impossibilité de le faire).

Une interface console pourrait être la suivante :

```
Entrez le problème : ra8 tb8a7 pb7 Rg6 De4 Ta1h8

   -----------------
8 | r t . . . . . T |    minuscules : noirs
7 | t p . . . . . . |    majuscules : blancs
6 | . . . . . . R . |    r R : roi
5 | . . . . . . . . |    d D : dame
4 | . . . . D . . . |    f F : fou
3 | . . . . . . . . |    c C : cavalier
2 | . . . . . . . . |    t T : tour
1 | T . . . . . . . |    p P : pion
   ------------------
    a b c d e f g h


Mat en 2 coups: possible avec l'unique solution De4-h1
```

Toute autre possibilité du logiciel dans le domaine du jeu d'échecs sera la bienvenue (y compris la généralisation au mat en n coups permettant de devenir champion du monde...)

Thème 10 Aide grammaticale

Le but de ce projet est d'écrire un ensemble de fonctions de grammaire, pouvant être intégré facilement dans tout programme de traitement de texte.

Les fonctions suivantes devront obligatoirement être disponibles :

- saisie d'un mot et vérification de son existence dans un dictionnaire ;
- proposition de mots proches en cas de non existence ;
- affichage du pluriel si c'est un nom ou un adjectif.

Facultativement, on pourra réfléchir à la manière de proposer les conjugaisons si c'est un verbe.

Thème 11 Recherches d'itinéraires en transport en commun

Le but de ce projet est de réaliser une version simplifié du service http://www.ratp.fr/itineraires/fr/ratp/recherche-avancee .

Le programme final demandera à l'utilisateur les stations de départ et d'arrivée et calculera et affichera l'itinéraire le plus rapide.

Dans un premier temps, on se bornera aux stations du RER intra-muros, en considérant que les durées de trajet inter-stations sont toutes égales, ainsi que les durées de correspondance.

Puis on intégrera les stations de métro intra-muros, et on utilisera des évaluations des durées de trajet et de correspondance.

Thème 12 Jeu de la vie

Le but de ce projet est de programmer le *Jeu de la vie*, simulation de l'évolution d'organismes en société :

Dans un damier rectangulaire de dimensions quelconques, des pions sont placés. Puis, à chaque génération, les pions apparaissent et/ou disparaissent sur le damier selon les règles suivantes :

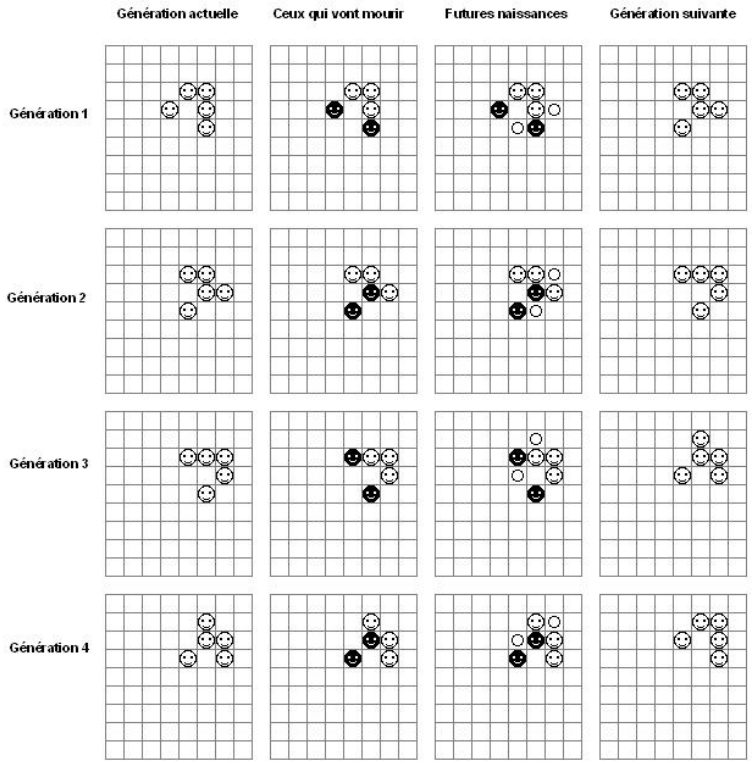
survie : tout pion voisin de deux ou trois pions survivra ;

naissance : toute case vide voisine de trois pions exactement donnera naissance à un pion ;

mort par étouffement : tout pion voisin de quatre pions ou plus disparaîtra ;

mort par isolement : tout pion isolé ou voisin d'un seul pion disparaîtra.

Ce sont les pions présents à chaque génération qui déterminent les naissances et les morts de la génération suivante. Par exemple, dans cette configuration :



Les principes de la simulation devront être étudiés, puis des choix d'interface utilisateur seront faits. Une version du jeu de la vie sera alors écrite.